

Global Positioning Logger

Senior Project

Matthew Hall
Computer Engineering Program
California Polytechnic State University
March 2010

1. Global Positioning Logger	1
• Senior Project	1
2. Introduction	3
• Statement of Problem	3
• Scope of the Work	3
• The Solution	4
3. Background.....	5
• Overview.....	5
• GPS Technology	5
• Microcontrollers/Arduino	6
4. Description	8
• Overview.....	8
• Brainstorming	8
• High-level Design	9
• Low-level Design	10
• Component 1: Speedometer.....	10
• Component 2: GPS Logging.....	11
• Component 3: GPS Mapping.....	11
• Purchasing	12
• Implementation.....	12
• Documentation.....	17
5. Evaluation	18
• Overview.....	18
• Testing	18
6. Related Works	21
• Overview.....	21
• Technical Research Resources	21
• Existing GPS Logger Products.....	21
7. Conclusion	23
• Project Experience	23
• Future Work	23
8. Bibliography	24
9. Appendix.....	25

Introduction

This written report is intended to fully describe and explain my Computer Engineering Senior Project, the "Global Positioning Logger" (GPL). This project was worked on during the Fall and Winter quarters of Cal Poly's 2009/2010 school year. Dr. Chris Lupo was the faculty advisor for this project. The GPL was completed to fulfill both the CPE461/462 curriculum requirements and to gain applicable solo-project experience. This final report will explain the reasons for the project, the technical details and an overall evaluation of the project's success and completion.

Statement of Problem

- To develop a mobile device that tracks the user's geographical location
- To develop a device that displays the user's real-time speed
- To gain further knowledge and experience in embedded system design and implementation

This project addresses the previously listed issues with a mobile embedded system. The GPL combines the use and knowledge of embedded hardware components and firmware code. This combination addresses the Cal Poly Computer Engineering Senior Project requirement that all "senior projects require the project to have both hardware and software components. [LUP09]"

Scope of the Work

The GPL was started at the beginning of Cal Poly's Fall 2009 quarter and was completed at the end of the Winter 2010 quarter. During the course of the project lifetime, several distinct phases were initiated and completed. The phases are as follows:

- Brainstorming
 - Creation and analysis of general project ideas/goals
 - Narrowing of possible project goals into a feasible product
- High-level design
 - General design of High-level components
 - Design of the interface between High-level components
- Low-level design
 - Internal design of High-level components
 - Decisions on specific hardware requirements
 - Design of software architecture
- Purchasing
 - Analysis of project budget
 - Research of hardware pricing
 - Hardware purchasing
- Implementation
 - Construction of hardware components
 - Creation of software files
- Testing
 - Testing of Global Positioning Satellite (GPS) logger
 - Testing of GPS speedometer
- Documentation
 - Creation of Senior Project Final Report
 - Review process with Senior Project Advisor

- **Note:** Informal progress documentation was created and updated throughout the project's entire life-cycle at www.hallmatthew.com

The Solution

The GPL addresses the Statement of Problem by creating a mobile device that utilizes GPS technology to record and display geo-locative information and speed.

The device:

- Can be hand-held, mounted on a bicycle, placed on a dashboard, carried in a backpack, etc.
- Provides the user with real-time feedback via an LCD and status LED
- Records GPS-derived location coordinates to an SD card
- Is mounted in a water-resistant and military spec case [PEL]
- Is powered by rechargeable AA batteries and has an average battery life of 12 hours

Background

Overview

This section of the report provides the technical details necessary to understand the technical implementation of the GPL. An overall knowledge of GPS technology and microcontrollers/Arduino is extremely helpful in understanding the engineering aspects of the GPL.

GPS Technology

In 1973, "Ivan Getting and Bradford Parkinson began leading a Defense Department project to provide continuous navigation information, leading to the development of GPS. The military launched the first GPS satellite in 1978 and completed the system in 1995. [JAM09]" This project has grown over the past few decades to include a "constellation" of twenty-four global positioning satellites that regularly transmit signals to Earth. Each satellite orbits the Earth twice a day. There are over thirty satellites in space, with the extras used as backups.

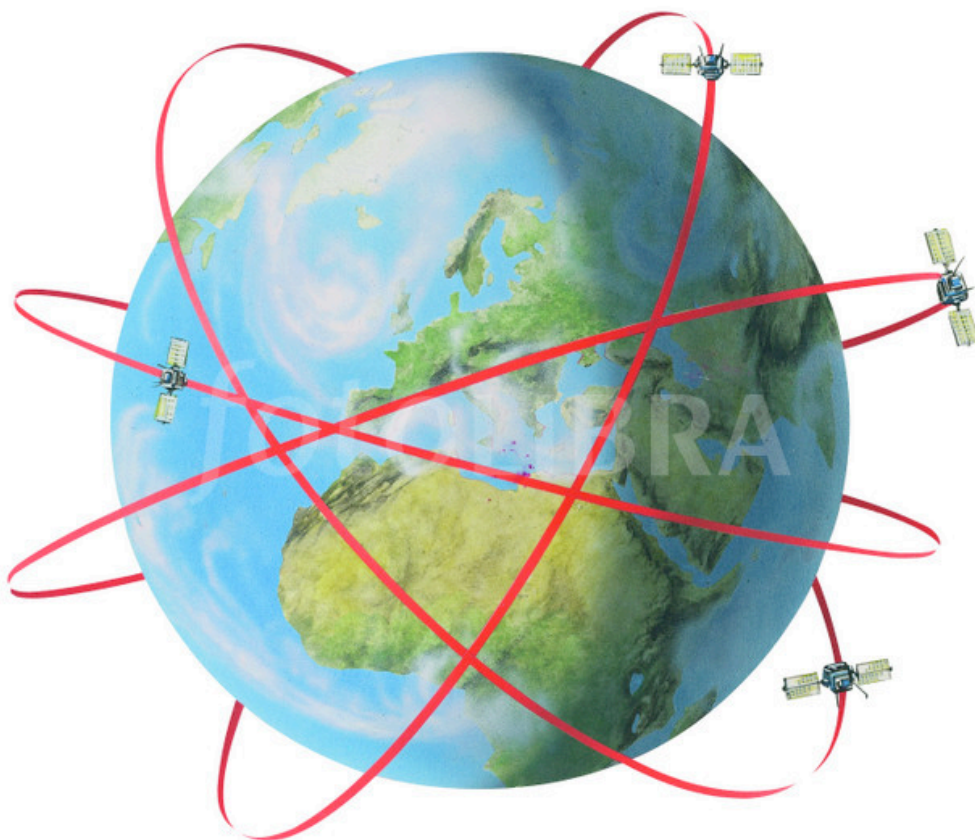


Figure 1: GPS constellation orbits [FOTOLIB]

The GPS system works in the following way:

- Satellites transmit a wireless signal to Earth encoded with current location and current time
- Multiple wireless signals as received by GPS receivers on the ground

- Differences in the signals are analyzed by the GPS receivers and data (speed, elevation, location, track angle, etc.) is calculated
- The calculated data is exported as formatted ASCII text, called an NMEA sentence

This data is used by the GPL to display the current time and speed. The data is also saved to a memory card for a visual display of past locations.

Microcontrollers/Arduino

The GPL's "brain" is a single-chip miniature computer. This device contains all of the basic components required be a self contained mini-computer, including a: CPU, clock, flash memory, RAM and I/O ports. It can run simple software programs (written in a high level language like C++) and can interface with other hardware components like LCDs. The microcontroller uses a very small amount of power and contains no moving parts (durable and reliable), thus making it an excellent choice for mobile device applications.

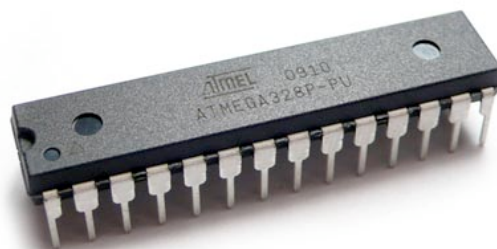


Figure 2: Atmel ATmega 328 chip [SEED]

The GPL uses an Arduino ("Are - Dwee - No") microcontroller development board. This is a small card-like device that houses an Atmel ATmega 328 microcontroller and all of the components necessary to support the microcontroller. The Arduino provides a more rugged and mobile solution than wiring the Atmel chip up to a breadboard. It also provides a USB connection to a PC which can be used for uploading code and communicating with the microcontroller at run-time.

The Arduino's software is written in a custom Arduino language, similar to C/C++. The code contains a minimum of two functions: the `setup()` and the `loop()`. In the `setup` any required initialization is performed. The code in the `loop` section is repeated over and over until the device is powered off.

There are many online development communities and tutorials that share code, wiring diagrams and tips on how to use the Arduino for different project applications. The openness and availability of software libraries and technical solutions allows people with little programming experience to develop very sophisticated embedded devices.

The Arduino board can also be mated to "Shields." These are boards of similar size that sit on top of the Arduino (interfaced via the I/O pins) and add additional functionality to the Arduino. There are Shields that provide ethernet connectivity, wireless point-to-point

communication, DC motor control, etc. The GPL uses the Adafruit GPS Logger Shield. This Shield provides GPS data connectivity and SD card read/write access.

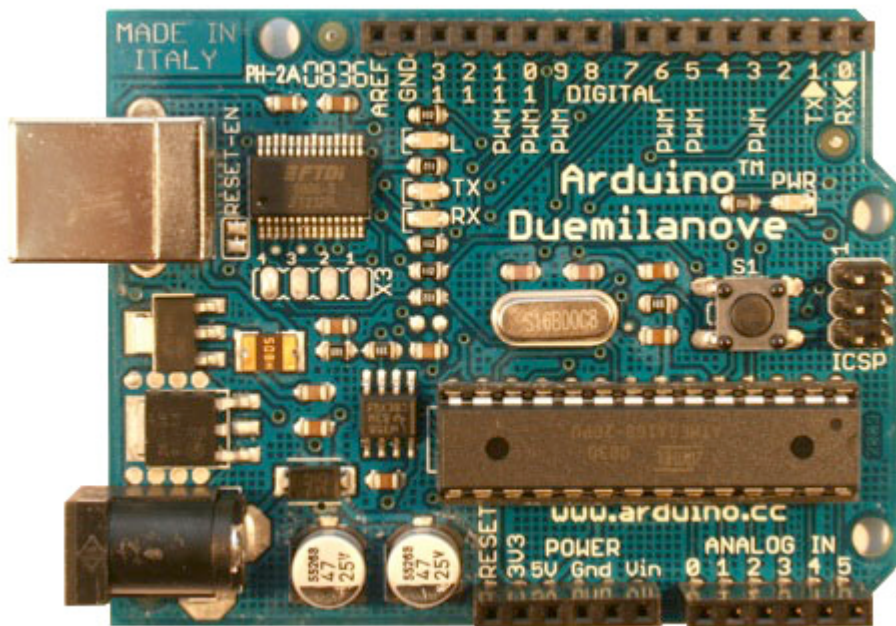


Figure 3: Arduino Duemilanove board [DUEM]

Description

Overview

This section of the Senior Project Report will address the technical progress and aspects of the GPL. An in-depth look will be taken into the design, implementation and testing phases of the project in order to gain a full understanding of how the GPL works.

As stated in the Introduction, the project progressed through the following phases of development:

- Brainstorming
- High-level design
- Low-level design
- Purchasing
- Implementation
- Testing (*covered in the Evaluation chapter of this report*)
- Formal documentation

The Description section of the report will examine each of these phases individually and expose the inner workings of the GPL project.

Brainstorming

The GPL was developed using a "top-down" design process. This process is a method of designing the High-level requirements first and then moving downward into the Low-level requirements. The GPL was brainstormed as a device that displays its speed and logs its location. From these requirements the lower-level specifications were developed (hardware/software choices, technologies used, etc.).

To design the GPL, I spoke to friends, colleagues and mentors in order to refine the project design into a clearer picture of what the product would be. The product was first envisioned to include a heart-rate monitor. This design was later refashioned to not include a heart-rate monitor. The main reasons for the design change were the additional costs of the hardware and increased development complexity.

At the end of the Brainstorming phase a solid overall product design was decided upon and the High-level component design interface was developed.

High-level Design

The High-level design phase was where the topmost project components were designed and the interfaces between them were designed. These components were the following:

1. Speedometer
2. GPS Logging
3. GPS Mapping

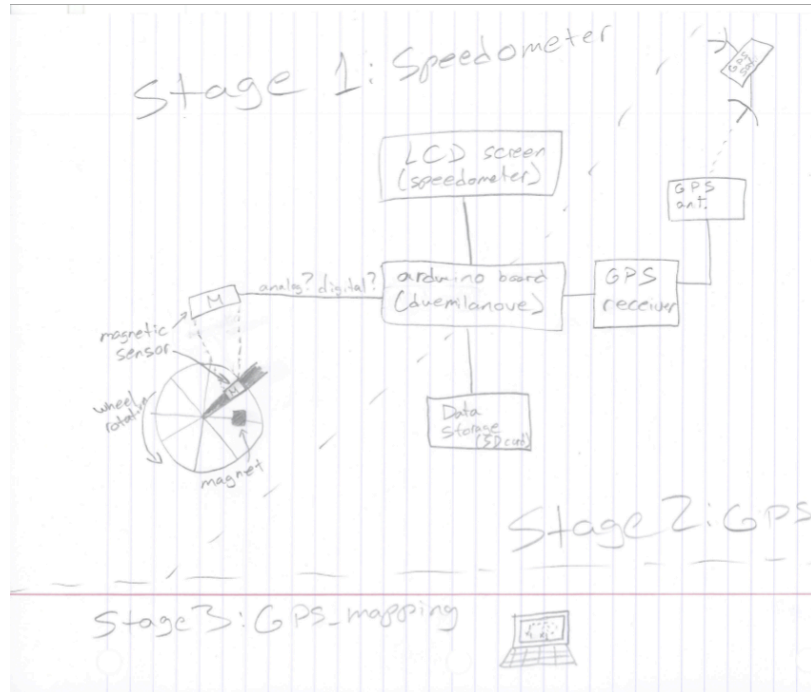


Figure 4: Original High-level design sketch

Figure 4, above, shows the original GPL High-level design components. The Speedometer (1) was originally designed to utilize a Reed switch to calculate bicycle speed. The GPS (2) design stayed the same throughout the entire project lifetime and the GPS_mapping (3) was later designed to use a free web application on the user's PC.

These three top level components were all designed to communicate with each other via the Arduino development board and the attached GPS Logger Shield.

Low-level Design

Once the High-level design phase was completed, the individual High-level components' inner workings had to be designed. At this point in the project's life-cycle, the design attitude was narrowed into a more technical mindset. The lower-level details had to be designed and decided upon, hardware specifications were made and the software flow design was built.

Component 1: Speedometer

The GPL's speedometer function design was changed midway through the Implementation process. Both the original and final designs will be presented here.

Original Design

At first, the GPL was designed to use a Reed switch to calculate the speed of a bicycle. A Reed switch is a small device that creates a closed switch when a magnetic field is next to the device. When there is no magnetic field next to the Reed switch it acts as an open switch circuit.

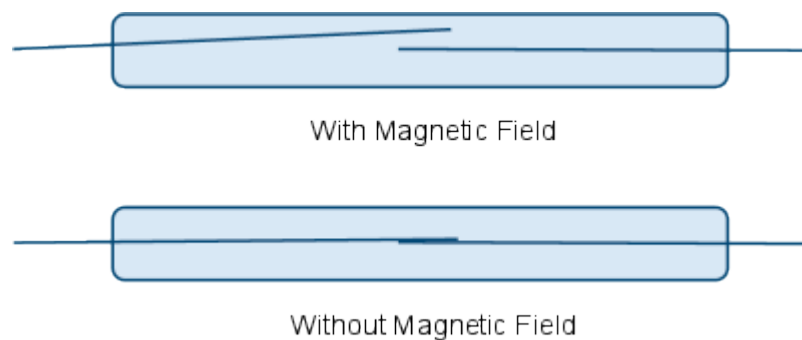


Figure 5: Reed Switch

The Reed switch was to be mounted on the front fork of a bicycle, with a magnet attached to a spoke of the bicycle's front wheel. Both were to be equidistant from the front axle, so as to pass by each other during the wheel's rotation.

This design would allow the Reed switch to close the circuit and the Arduino could sense a voltage on the line that was previously at 0V. Using this hardware system, the software would calculate the time elapsed between complete wheel revolutions. This value would be saved as Δt .

The distance traveled by the bicycle (Δd) during Δt is equal to the circumference of the bicycle tire. This circumference is found using the following equation: $\text{Circumference} = 2 * \pi * \text{Radius}$. Using this information, the speed of the bicycle could be calculated using $\text{Speed} = \Delta d / \Delta t$.

Final Design

During the Implementation phase, it was discovered that the GPS module calculated its own speed with great accuracy. The Reed switch based Speedometer was then removed from the project.

Using the GPS module's calculated speed, the GPL would be a far more versatile product. It would no longer be physically bound to a bicycle by wires, creating a more portable product. The GPL could also be used by a pedestrian, motorist, pilot, etc. The lack of

reliance on bicycle wheel revolution allowed the GPL to be used in more situations than its predecessor's design.

The speed data is calculated by the GPS module and placed in an NMEA sentence. This sentence is read by the Arduino and displayed on LCD.

Component 2: GPS Logging

The GPS logging capabilities of the GPL were provided by the Adafruit GPS Logger Shield. This device attached to the Arduino's I/O pins and allowed the system to:

- Obtain GPS data
- Read/Write data to an SD card

With these two capabilities, a log of every captured NMEA sentence could be created using the following algorithm:

As an NMEA sentence is captured by the Arduino (sent by the GPS module via a serial communication buffer), the entire sentence was saved to the SD card. As time progresses, the log grows and displays a history of the device's GPS coordinates, speed, elevation, etc.

In order to create intuitive and easy to understand logs, every time the GPL is power cycled a new log file is created on the SD card. This removes log analysis confusion by introducing modularity into the design and use.

Component 3: GPS Mapping

A major piece of the GPL's functionality comes from its ability to display its GPS log files in a useful way. Therefore, the design goal was to translate the log file data (from Component 2) into a Google Map. The Google Map would display the route that the GPL took during its run-time.

Google provides APIs to create Google Map applications. The original design for component 3 was to use the APIs to create a custom application that read a log file, as input, and produced a Google Map with the GPL's route, as output. This design was then dropped when GPSVisualizer.com was discovered to do the exact same thing. This website can read plain text files, containing raw NMEA ASCII sentences and produce the corresponding Google Map.

Purchasing

Once the High-level and Low-level phases were complete, the hardware could be purchased. The previous two design phases had to be completed before buying the hardware, lest the designs change and the wrong materials be purchased.

The first step in the Purchasing phase was to produce an itemized list of hardware. This list included the required project hardware, item descriptions, quantities and associated costs. The list was produced using spreadsheet software. The spreadsheet can be found in the appendix.

After the initial itemized hardware list was compiled, items were purchased. As items were purchased, a record of purchase was documented on the spreadsheet. The Implementation phase overlaps the Purchasing phase, as parts were put together and software was written before all of the hardware arrived.

Implementation

GPS Hardware

The Implementation phase started with the physical construction of the GPS Shield and battery pack. These pieces of hardware were both purchased from Adafruit Industries and come in kits. The kits include a PCB and a bag of loose electronic components. The components were soldered onto the PCBs and tested as the manufacturer recommends.

Once the GPS shield was put together, it was interfaced with the Arduino board. This Arduino/GPS mapping module was then tested with a PC (via serial port) to ensure the GPS module was receiving GPS data and producing legitimate data. The serial port log displayed the raw NMEA sentences coming from the GPS module. These sentences were parsed manually and the longitude/latitude coordinates were checked using Google Maps. The module worked during the first test, displaying results around thirty meters from its actual location.

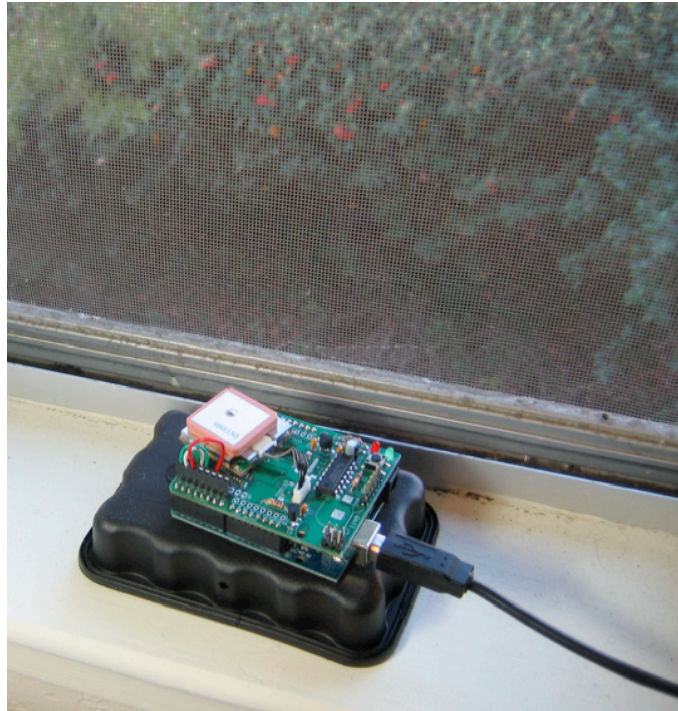


Figure 6: GPS module testing on windowsill

```

/dev/tty.usbserial-A6008sDs
Send

SGPGSV,3,1,12,14,87,076,,26,57,044,34,22,55,045,30,19,40,272,34*7C
SGPGSV,3,2,12,06,32,224,18,03,31,236,27,18,24,072,,21,15,132,*7F
SGPGSV,3,3,12,31,12,169,,09,11,041,33,32,08,272,32,27,06,035,*73
SGPRMC,004015.000,A,3517.8812,N,12040.5425,W,0.41,347.63,090110,,*1A
SGPGGA,004016.000,3517.8811,N,12040.5425,W,1.07,1.8,93.8,M,-30.6,M,0000*58
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004016.000,A,3517.8811,N,12040.5425,W,0.13,157.52,090110,,*1C
SGPGGA,004017.000,3517.8810,N,12040.5425,W,1.07,1.8,93.5,M,-30.6,M,0000*55
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004017.000,A,3517.8810,N,12040.5425,W,0.59,155.21,090110,,*14
SGPGGA,004018.000,3517.8808,N,12040.5425,W,1.07,1.8,93.2,M,-30.6,M,0000*54
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004018.000,A,3517.8808,N,12040.5425,W,0.19,145.90,090110,,*1D
SGPGGA,004019.000,3517.8809,N,12040.5426,W,1.07,1.8,93.3,M,-30.6,M,0000*56
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004019.000,A,3517.8809,N,12040.5426,W,0.21,39.61,090110,,*21
SGPGGA,004020.000,3517.8809,N,12040.5426,W,1.07,1.8,93.2,M,-30.6,M,0000*5D
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPGSV,3,1,12,14,87,076,,26,57,044,34,22,55,045,30,19,40,272,35*7D
SGPGSV,3,2,12,06,32,224,19,03,31,236,28,18,24,072,,21,15,132,*71
SGPGSV,3,3,12,31,12,169,,09,11,041,33,32,08,272,32,27,06,035,*73
SGPRMC,004020.000,A,3517.8809,N,12040.5426,W,0.24,20.22,090110,,*21
SGPGGA,004021.000,3517.8809,N,12040.5426,W,1.07,1.8,93.2,M,-30.6,M,0000*5C
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004021.000,A,3517.8809,N,12040.5426,W,0.22,35.05,090110,,*17
SGPGGA,004022.000,3517.8808,N,12040.5426,W,1.07,1.8,92.9,M,-30.6,M,0000*54
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004022.000,A,3517.8808,N,12040.5426,W,0.21,159.14,090110,,*1D
SGPGGA,004023.000,3517.8807,N,12040.5427,W,1.07,1.8,92.7,M,-30.6,M,0000*55
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004023.000,A,3517.8807,N,12040.5427,W,0.05,26.91,090110,,*20
SGPGGA,004024.000,3517.8807,N,12040.5428,W,1.07,1.8,92.4,M,-30.6,M,0000*5E
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004024.000,A,3517.8807,N,12040.5428,W,0.06,13.32,090110,,*24
SGPGGA,004025.000,3517.8806,N,12040.5428,W,1.07,1.8,92.1,M,-30.6,M,0000*58
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPGSV,3,1,12,14,87,076,,26,57,044,35,22,55,045,30,19,40,272,35*7C
SGPGSV,3,2,12,06,32,224,21,03,31,236,28,18,24,072,,21,15,132,*7A
SGPGSV,3,3,12,31,12,169,,09,11,041,34,32,08,272,31,27,06,035,*77
SGPRMC,004025.000,A,3517.8806,N,12040.5428,W,0.09,140.26,090110,,*19
SGPGGA,004026.000,3517.8805,N,12040.5429,W,1.07,1.8,91.8,M,-30.6,M,0000*50
SGPGSA,A,3,06,03,26,19,22,32,09,,,,,2.5,1.8,1.7*38
SGPRMC,004026.000,A,3517.8805,N,12040.5429,W,0.13,148.00,090110,,*1F
SGPGGA,004027.000,3517.8804,N,12040.5428,W,1.07,1.8,91.8,M,-30.6,M,0000*51

```

Figure 7: GPS module output serial log

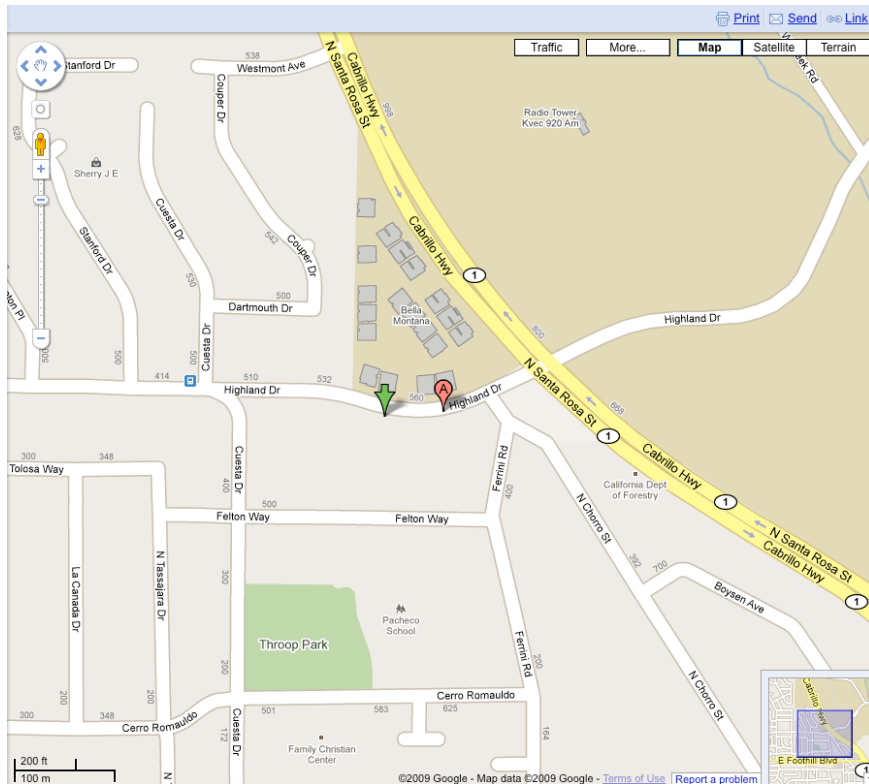


Figure 8: GPS module coordinates (A) vs. actual location (Arrow)

GPS Logging Software

After the initial GPS module tests were completed, the logging software was written. The logging code that runs on the Arduino's microcontroller follows a simple algorithm. The code runs through an infinite loop, checks for updated GPS data and writes the new data to the SD card.

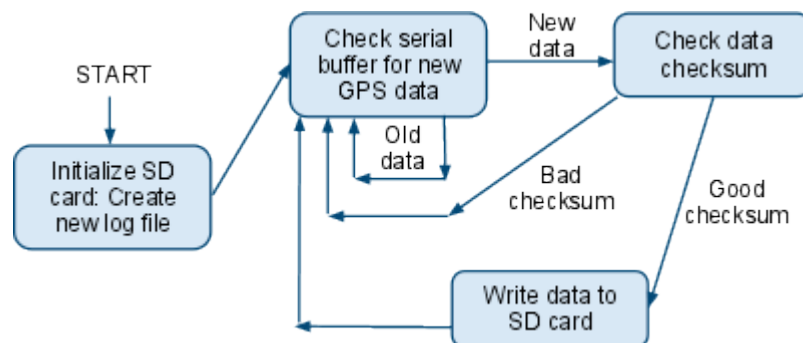


Figure 9: GPS logging software

Real-time Display

After the logging component of the software was complete, the real-time features of the GPL were implemented. The real-time features were narrowed down to the following pieces of data, displayed on the LCD:

- Current time (PST)
- Current speed (mph)
- GPS link status (Void or Active)

In order to display these three data fields, the LCD had to be interfaced with the Arduino. This was first accomplished using a breadboard and the Arduino LiquidCrystal 4-bit library [LIQ09]. This library provides an API for use with most 16x2 character

LCDs.

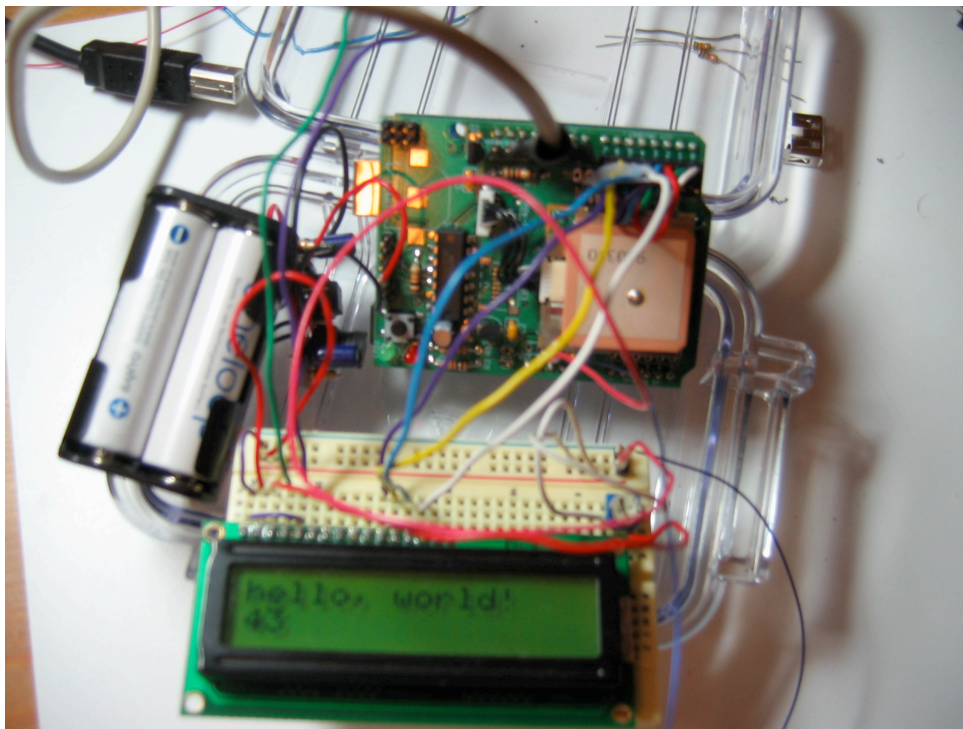


Figure 10: LCD interfaced with Arduino via breadboard

After the LCD was successfully interfaced with the Arduino, with a breadboard, the final software implementation was coded. The final solution merged the LCD functionality with the GPS logging capabilities. This conglomeration of features allowed the GPL to accomplish its primary objectives:

- To develop a mobile device that tracks the user's geographical location
- To develop a device that displays the user's real-time speed
- To gain further knowledge and experience in embedded system design and implementation

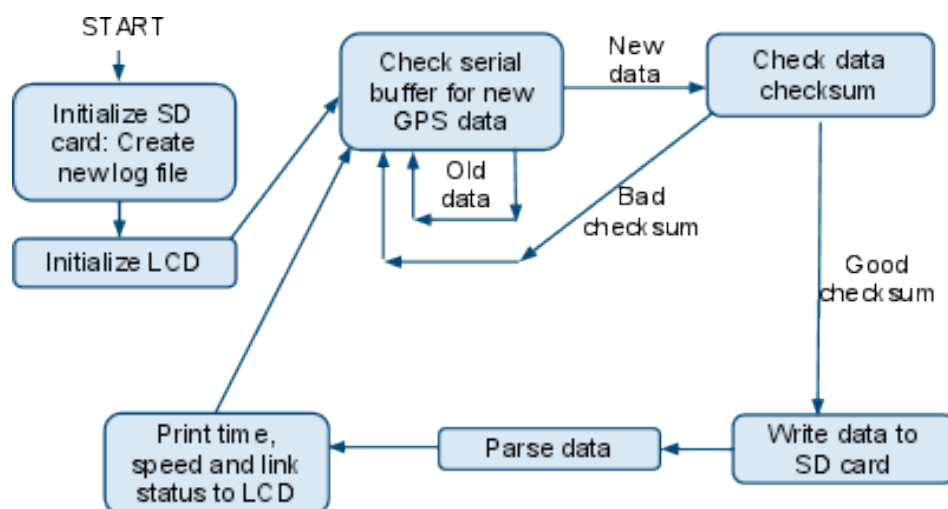


Figure 11: Complete software architecture

Final Product Packaging

At this point in the project, the GPL was fully functional, albeit not very portable. The GPL needed a more refined package solution in order to be useful in real-world applications. The loose wiring, delicate breadboard and exposed metal contacts created durability issues that needed to be addressed.

The GPL, according to the original design specifications and itemized hardware list, was to be packaged in a Pelican Case. This case is a rugged, water-resistant and hardened plastic box that meets MIL-STD 4150-J durability standards.

In order to package the GPL into the Pelican Case, the breadboard assembly needed to be removed. The breadboard was replaced by a custom built wiring harness with plug and play functionality. This wiring harness allows the LCD to be plugged into the Arduino/GPS shield very easily. It also allows multiple LCDs to be swapped in and out, creating a more modular system, for increased robustness.

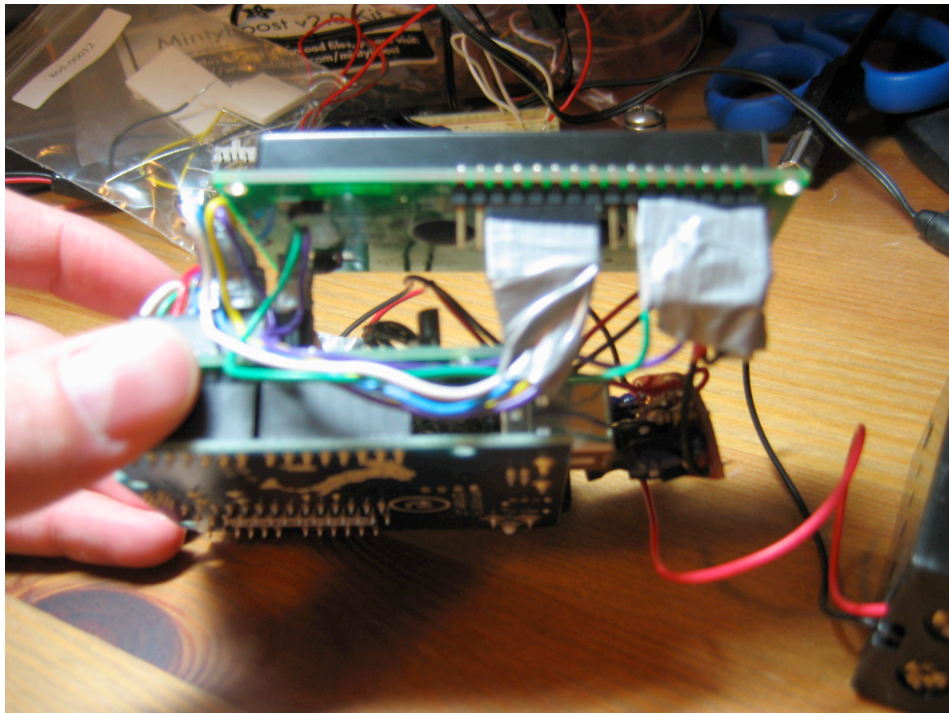


Figure 12: LCD wiring harness

Upon completion of the wiring harness, the entire GPL system was placed in the Pelican Case for ease of use and transport. At this point, the GPL was completely assembled and ready for final testing.



Figure 13: Final GPL packaging

Documentation

The GPL project was accompanied by two main forms of documentation, the ongoing progress blog and the final report. The blog was kept on a website (www.hallmatthew.com) and updated as the project progressed. The final report was written and compiled at the tail end of the GPL project.

Blog

The GPL's development cycle went through several different stages. Some stages were more hardware oriented, some were software oriented. To create a transparent development history, the GPL blog was updated periodically with posts showing the most recent progress of the project. Text, pictures and videos were included in the posts to give readers a full understanding of the project and its components.

Final Report

After the GPL was physically assembled and tested, the final report was started. This document was created to provide a more formal description of the project than the blog.

Evaluation

Overview

The Evaluation section of this report describes the tests performed on the GPL. These tests were created and performed in order to evaluate the successes/failures of the GPL.

Testing

After the GPL was fully assembled and packaged, the Testing phase was begun. This phase involved the following tests:

- **Automobile** speedometer comparison
- **Automobile** GPS logging review
- **Outdoor** walking speed review
- **Outdoor** walking GPS logging review
- **Indoor** walking speed review
- **Indoor** walking GPS logging review

During these tests, the device was powered on, GPS data was logged and the indicated speed was recorded.

The tests were performed in order to evaluate the following criteria:

- Indicated speed accuracy
- Indicated location accuracy

These pieces of information were evaluated with the following metrics:

- Automobile
 - Speed Pass: $\leq \pm 2$ mph compared to vehicle speedometer
 - Location Pass: Indicated route follows actual route
- Outdoor walking
 - Speed Pass: Indicated speed of ~ 3 mph
 - Location pass: Indicated route follows actual route
- Indoor walking
 - Speed Pass: Indicated speed of ~ 3 mph
 - Location pass: Indicated route follows actual route

Note: GPS calculated speed is more accurate at higher speeds, therefore the indicated speeds while stopped are not assumed to be accurate and will not pass these tests. Therefore, the indicated speeds, while stopped, will not be evaluated by the PASS/FAIL metrics.

Automobile

The GPL was used during an automobile trip in San Luis Obispo, CA. The GPL's indicated speed was compared to the automobile's speedometer. The results showed that both speeds were identical (± 1 mph) except during a stop, where the automobile's speed was 0 mph and the GPL indicated speeds between 0 and 1 mph.

After the speed test was completed, the GPS log was analyzed using GPSVisualizer.com.

The resulting map showed an almost perfect route taken by the automobile.

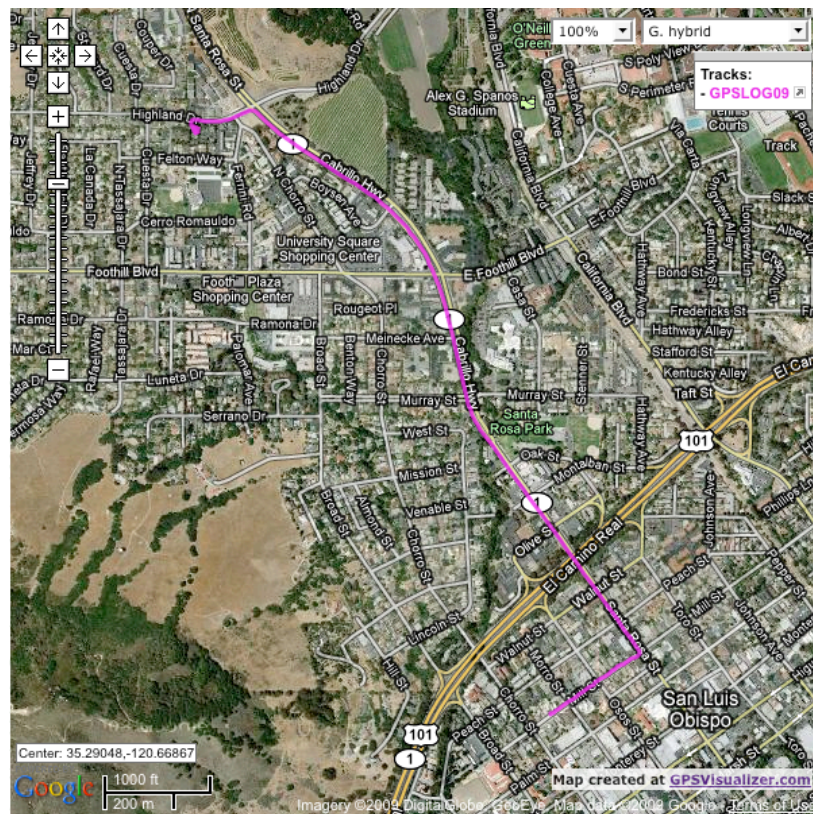


Figure 14: Automobile GPS logging route

These tests indicate passing scores for both the speed and location capabilities at automobile freeway speeds (~65 mph).

Outdoor Walking

This test involved walking with the GPL in a backpack to test the low-speed functionality of the project. The results, when at a stop, were the same as the stopped automobile tests. When walking, the GPL indicated an average speed of ~3 mph, which is a reasonably accurate walking speed.

The GPS logging results were just as accurate as the automobile GPS logging test results and displayed a very accurate route taken by the tester.

Both of the outdoor tests were passed by the GPL.

Indoor Walking

The indoor tests revealed a weakness in the GPL's capabilities. Due to a limitation of GPS technology, the GPL's indicated speed and location is not very accurate while indoors. This is assumed to be caused by the building materials, which block GPS signals and reduce the signal strength.

Again, while standing still the GPL indicated a range of speeds from 0 to 2 mph. While walking indoors, the indicated speed was ~3 mph.

The indoor GPS location logging was very inaccurate, often reporting the device to be in a neighboring building or across the street.

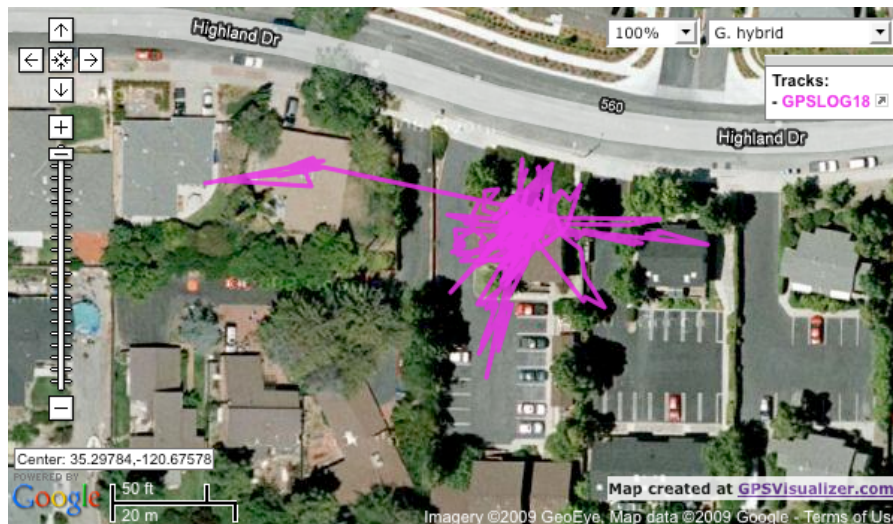


Figure 15: Indoor GPS logging accuracy

The indoor tests were split with a passing score on speed but a failing score on location logging.

Testing Results

The following test results were obtained using the test metrics:

- Automobile
 - Speed: **PASS**
 - Location: **PASS**
- Outdoor walking
 - Speed: **PASS**
 - Location: **PASS**
- Indoor walking
 - Speed: **PASS**
 - Location: **FAIL**

These results show that the GPL passed 5 out of 6 tests, giving it an overall grade of ~83% passing. Had the tests included testing the indicated speed, while stopped, then the GPL would not have done as well. The overall reasons for any failure or inaccuracies found in the GPL are a result of the underlying technologies used by the GPL, not flaws in the GPL's design.

Related Works

Overview

A number of projects were referenced, studied and learned from to design and implement the GPL. This section of the report lists and details projects with similar technologies, components and objectives as the GPL.

Technical Research Resources

Arduino Communities- One major contributor to the GPL is the Arduino development community. This community of engineers, artists and hobbyists constantly thinks up new uses for the Arduino development board. Online forum communities at www.arduino.cc and www.adafruit.com were used for the majority of technical research. Members of these communities work on Arduino-based projects and post questions, research, results and tutorials on the websites.

Examples of existing Arduino community projects (from which information, code and hardware designs were gleaned from) include the following:

- LiquidCrystal 4-bit library [LIQ09]
 - The LiquidCrystal 4-bit Arduino software library is a compilation of C/C++ code to control a generic 16x2 character LCD. This library provides an API for initializing, printing, clearing and formatting an LCD with a standard HD44780 chipset.
 - Included with the library documentation are schematics which display recommended wiring schemes
- NewSoftSerial library [NEW09]
 - This Arduino library provides an API for serial communication to serial-capable devices. The SP's uses this library to communicate with the EM-406A GPS module.
- AF_SDlog library [AFS09]
 - This library allows read and write access to an SD card located on the GPS shield

Adafruit/Ladyada GPS logger project- The Adafruit/Ladyada GPS logger project provided a large base of knowledge for the GPL [ADA09]. This project is run by Lemor Fried of www.adafruit.com. This GPS logger project distributes hardware, software and tutorials to create the Adafruit GPS shield and interface it with an Arduino development board. The GPS shield is an integral component to the GPL as it provides the GPS logging and SD card functionality.

Existing GPS Logger Products

Several commercial GPS loggers are available for purchase. These products provide much of the same functionality as the GPL. Many do so in a smaller package than the GPL, and for a lower price.

Some of these products were examined in order to develop and narrow a high-level design idea for the SP. The following products were investigated:

- Arduino GPS module from Libellium [LIB08]

- This project was of interest because of its use of the same GPS module chipset that the GPL uses (GlobalSat EM-406A). This project provided wiring diagrams and open source software that was studied and gleaned from.
- OHARARP waterproof GPS logger [OHA09]
 - This GPS project provided the idea to house the GPL in a rugged and water-resistant casing. The OHARARP GPS logger and GPL are both contained in Pelican Cases, although the GPL uses a smaller version case for a smaller form factor.
- Sparkfun Digital Speedometer [SPA09]
 - The Sparkfun Digital Speedometer kit provided the inspiration for real-time data display on the GPL's LCD. This project also provides sample hardware configurations and software samples to learn from.

Conclusion

Project Experience

This Senior Project was a great chance to put the Cal Poly slogan into action. "Learn by Doing" stresses the importance of putting knowledge to work, in order to gain new knowledge. This is exactly what was accomplished during the GPL project.

As a fifth-year Computer Engineering undergraduate, I have gained a wealth of technical knowledge. My topics of knowledge include embedded systems, engineering economics, project management, and good design principles. All of these topics were used to build the GPL and all of these areas of knowledge were added to and built up during the lifetime of the project. This is what made the Senior Project so worthwhile and successful, it was a learning experience in and of itself. The Cal Poly Computer Engineering Senior Project is a vital component to the College of Engineering's curriculum and has given me a better understanding of technical projects and has made me a better engineer.

Future Work

Due to the limited time-frame that the GPL had to be completed during, the project experienced some cuts to its final implementation. If the GPL is to be worked on again in the future, the modifications should be to the physical packaging and the software.

Physical Packaging Improvements

The GPL, while packed into a durable case, is not as portable and robust as it could be. There are still exposed wires and contacts that can be touched when the case is open. The power switch is delicate and extra care is taken when switching it, to avoid breaking the leads attached to it. Also, the wiring harness for the LCD could use some professional touches. These changes would include removing the duct tape insulation and applying heat shrink insulation. Overall, a general physical "polish" should be applied to the wiring of the project to make it less delicate.

Software Improvements

The GPL firmware, while efficient and usable, could use more in-depth testing of the SD card. This would ensure that errors are caught correctly and dealt with properly. The GPL has worked well, but testing for non-typical usage scenarios was not performed. The following scenarios could be run in the future:

- No SD card
- Incorrectly formatted SD card
- Full SD card
- Damaged SD card

Bibliography

[LUP09] Lupo, Chris. "Senior Project Guidelines." San Luis Obispo, United States of America, December, 2009.

[PEL] Pelican Products Website. <http://www.pelican.com/support/faq.php/>.

[JAM09] James, Randy. "A Brief History of GPS." *Time*, May 26, 2009.

[FOTOLIB] fotoLIBRA Website. "Diagram Showing How GPS Works." <http://gb.fotolibra.com/images/previews/49528-diagram-showing-how-gps-works-illustratio.jpeg/>.

[SEED] SEEED DEPOT Website. "328." <http://www.seeedstudio.com/depot/images/product/328p.jpg/>.

[DUEM] Arduinio Duemilanove Website. "ArduinoDeumilanove." <http://arduino.cc/en/uploads/Main/ArduinoDuemilanove.jpg/>.

[LIQ09] Official Arduino Website. <http://www.arduino.cc/en/Tutorial/LiquidCrystal/>.

[NEW09] Arduiniana Blog, The. <http://arduiniana.org/libraries/NewSoftSerial/>.

[AFS09] Ladyada Website. http://www.ladyada.net/make/gpsshield/AF_SDLog%20v2.zip/.

[ADA09] Ladyada Website. <http://www.ladyada.net/make/gpsshield/index.html/>.

[LIB08] Libelium Wireless Distributed Communications Website. <http://www.sensor-networks.org/index.php?page=0816816801>.

[OHA09] OHARARP LLC. Website. <http://www.ohararp.com/products.html/>.

[SPA09] Sparkfun Electronics Tutorials. http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=123/.

Appendix

GPS/Speedometer Project Parts List			Tax Rate: 1.08				
Item Name	Quantity	Unit Price	Shipping Price	Total Part Price (incl tax)	Vendor	Purchase	Item Description
Arduino Duemilanove Board	1	\$29.00	\$4.49	\$35.81	Amazon	1	Microcontroller board base unit
Arduino GPS logger shield v1.0	1	\$19.50	\$4.67	\$25.73	Adafruit	1	Add-on shield to interface with GPS module
EM-406A GPS Module	1	\$59.95	\$8.88	\$73.63	Adafruit	1	SIRF III Receiver w Antenna
LCD Display	1	\$0.00	\$0.00	\$0.00	N/A	1	Common 16x2 character lcd w/ 16 pin interface
Reed Switch	3	\$1.50	\$3.22	\$14.52	Sparkfun	1	Speedometer pulse switch. 2 position switch.
Magnet Cube	1	\$1.50	\$3.22	\$4.84	Sparkfun	1	Rare earth magnet to engage reed switch. 0.25" cubed.
SD Card	1	\$6.95	\$2.99	\$10.50	Newegg	1	Kingston 2GBET. 2 GB flash memory card.
Pelican Micro Case 1010	1	\$13.95	\$0.00	\$15.07	Amazon	1	Rugged, water-resistant case. 149X103x54mm ext. 111x73x43mm int.
Minty Boost Battery Pack Kit	1	\$19.50	\$3.82	\$24.88	Adafruit	1	3V → 5V USB charger circuitry. Provides power to arduino.
Sanyo Eneloop AA charger w/ batteries	1	\$17.25	\$0.00	\$18.63	Amazon	1	Charger with 4 2000mAh AA batteries
Altoids Gum Tin	1	\$2.00	\$0.00	\$2.16	Local Grocer	1	Minty Boost charger housing
Liquid Electrical Tape	1	\$8.00	\$0.00	\$8.64	Radio Shack	1	Seal pelican box wire-thru holes
Pelican foam	1	\$4.50	\$0.00	\$4.86	Amazon	1	Foam for pelican box. Pluck and pick style.
			Project Cost:	\$234.40			

Table 1: Itemized parts list